

Generalized Pattern Search Algorithms with Adaptive Precision Function Evaluations¹

Elijah Polak² and Michael Wetter³

May 14, 2003

Abstract

In the literature on generalized pattern search algorithms, convergence to a stationary point of a once continuously differentiable cost function is established under the assumption that the cost function can be evaluated exactly. However, there is a large class of engineering problems where the numerical evaluation of the cost function involves the solution of systems of differential algebraic equations. Since the termination criteria of the numerical solvers often depend on the design parameters, computer code for solving these systems usually defines a numerical approximation to the cost function that is discontinuous with respect to the design parameters. Standard generalized pattern search algorithms have been applied heuristically to such problems, but no convergence properties have been stated.

In this paper we extend a class of generalized pattern search algorithms to a form that uses adaptive precision approximations to the cost function. These numerical approximations need not define a continuous function. Our algorithms can be used for solving linearly constrained problems with cost functions that are at least locally Lipschitz continuous.

Assuming that the cost function is smooth, we prove that our algorithms converge to a stationary point. Under the weaker assumption that the cost function is only locally Lipschitz continuous, we show that our algorithms converge to points at which the Clarke generalized directional derivatives are nonnegative in predefined directions.

An important feature of our adaptive precision scheme is the use of coarse approximations in the early iterations, with the approximation precision controlled by a test. Such an approach leads to substantial time savings in minimizing computationally expensive functions.

Key words: Algorithm implementation, approximations, generalized pattern search, Hooke-Jeeves, Clarke's generalized directional derivative, nonsmooth optimization.

¹This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy Office of Building Technology, State and Community Programs, Office of Building Research and Standards, of the U.S. Dept. of Energy, under Contract No. DE-AC03-76SF00098 and the National Science Foundation under Grant No. ECS-9900985.

²Department of Electrical Engineering, University of California at Berkeley, Berkeley, CA 94720, USA (polak@eecs.berkeley.edu).

³Simulation Research Group, Building Technologies Department, Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA (MWetter@lbl.gov).

1 Introduction

Generalized pattern search (GPS) algorithms are derivative free methods for the minimization of smooth functions, possibly with linear inequality constraints. Examples of pattern search algorithms are the coordinate search algorithm [12], the pattern search algorithm of Hooke and Jeeves [8], and the multidirectional search algorithm of Dennis and Torczon [6]. What they all have in common is that they define the construction of a mesh, which is then explored according to some rule, and if no decrease in cost is obtained on mesh points around the current iterate, then the mesh is refined and the process is repeated.

In 1997, Torczon [15] was the first to show that all the existing pattern search algorithms are specific implementations of an abstract pattern search scheme and to establish that for unconstrained problems with smooth cost functions, the gradient of the cost function vanishes at accumulation points of sequences constructed by this scheme. Lewis and Torczon extended her theory to address bound constrained problems [9] and problems with linear inequality constraints [10]. In both cases, convergence to a feasible point x^* satisfying $\langle \nabla f(x^*), x - x^* \rangle \geq 0$ for all feasible x is proven under the condition that $f(\cdot)$ is once continuously differentiable. Audet and Dennis [1] present a simpler abstraction of GPS algorithms, and, in addition to reestablishing the Torczon and the Lewis and Torczon results, they relax the assumption that the cost function is smooth to that it is locally Lipschitz continuous. However, their characterization of accumulation points of sequences constructed by a GPS algorithm, on a locally Lipschitz continuous cost function, while not without merit, falls short of showing that the accumulation points are stationary in the Clarke sense [3] (i.e., $0 \in \partial^0 f(x^*)$). It does not seem possible to improve their result.

In principle, a natural area for the application of GPS algorithms is engineering optimization, where the cost functions are defined on the solution of complex systems of equations including implicit equations, ordinary differential equations, and partial differential equations. However, in such cases, obtaining an accurate approximation to the cost function often takes many hours, and there is no straightforward way of approximating gradients. Furthermore, it is not uncommon that the termination criteria of the numerical solvers introduce discontinuities in the approximations to the cost function. Hence, standard GPS algorithms can only be used heuristically in this context.

Even if it were possible to characterize numerical approximation errors as random noise, it follows from [17] that obtaining a reasonably accurate solution would involve, eventually, a prohibitively large number of function evaluations per iteration. Therefore, attempting to characterize numerical errors as random noise does not appear to be a promising approach in the context of solving major classes of engineering optimization problems by GPS algorithms.

In this paper we present a modified class of GPS algorithms which adjust the precision of the function evaluations adaptively: low precision in the early iterations, with precision progressively increasing as a solution is approached. The modified GPS algorithms converge to stationary points of the cost function even though the cost function is approximated by a family of discontinuous functions.

The GPS algorithms that we present are somewhat simpler in structure than those

presented in [15, 9, 10, 1]. We assume that the cost function $f(\cdot)$ is at least locally Lipschitz continuous and that it can be approximated by a family of functions, say $\{f_N(\cdot)\}_{N \in \mathbb{N}^q}$ with fixed $q \in \mathbb{N}$, where each $f_N(\cdot)$ may be discontinuous but converges to $f(\cdot)$ uniformly on bounded sets. A test in the algorithm determines when precision must be increased. This test makes use only of the current mesh size and includes parameters that can be used to control the speed with which precision is increased. This flexibility can be exploited to obtain an order of magnitude reduction in computing times, as compared to using high precision throughout the computation. Since our GPS algorithms include global search and local search stages, as is typical in GPS algorithms, our GPS algorithms can also be used with surrogate cost functions for the global search, as in [7, 16, 14, 2].

Under the assumption that the cost function is continuously differentiable, all the accumulation points constructed by our GPS algorithms are stationary, while under the assumption that $f(\cdot)$ is only locally Lipschitz continuous, our algorithms converge to points at which the Clarke generalized directional derivatives are nonnegative in predefined directions. Thus, we regain the results of [1].

2 Notation

1. We denote by \mathbb{Z} the set of integers, by \mathbb{Q} the set of rational numbers, and by $\mathbb{N} \triangleq \{0, 1, \dots\}$ the set of natural numbers. The set \mathbb{N}_+ is defined as $\mathbb{N}_+ \triangleq \{1, 2, \dots\}$. Similarly, vectors in \mathbb{R}^n with strictly positive elements are denoted by $\mathbb{R}_+^n \triangleq \{x \in \mathbb{R}^n \mid x^i > 0, \forall i = 1, \dots, n\}$ and the set \mathbb{Q}_+ is defined as $\mathbb{Q}_+ \triangleq \{q \in \mathbb{Q} \mid q > 0\}$.
2. The inner product in \mathbb{R}^n is denoted by $\langle \cdot, \cdot \rangle$ and for $x, y \in \mathbb{R}^n$ defined by $\langle x, y \rangle \triangleq \sum_{i=1}^n x^i y^i$.
3. For $N \in \mathbb{N}^q$, by $N \rightarrow \infty$, we mean that each component of N tends to infinity.
4. If a subsequence $\{x_i\}_{i \in \mathbf{K}} \subset \{x_i\}_{i=0}^\infty$ converges to some point x , we write $x_i \rightarrow^{\mathbf{K}} x$.
5. Let \mathbb{W} be a set containing a sequence $\{w_i\}_{i=0}^k$. Then, we denote by \underline{w}_k the sequence $\{w_i\}_{i=0}^k$ and by $\underline{\mathbf{W}}_k$ the set of all $k+1$ element sequences in \mathbb{W} .
6. We denote by $\{e_i\}_{i=1}^n$ the unit vectors in \mathbb{R}^n .
7. If \mathbf{X} is a set, we denote by $\partial \mathbf{X}$ its boundary and by $\overline{\mathbf{X}}$ its closure.
8. If \mathbf{S} is a set, we denote by $2^{\mathbf{S}}$ the set of all nonempty subsets of \mathbf{S} .
9. If $\widehat{D} \in \mathbb{Q}^{n \times q}$ is a matrix, we will use the notation $\widehat{d} \in \widehat{D}$ to denote the fact that $\widehat{d} \in \mathbb{Q}^n$ is a column vector of the matrix \widehat{D} . Similarly, by $D \subset \widehat{D}$ we mean that $D \in \mathbb{Q}^{n \times p}$ ($1 \leq p \leq q$) is a matrix containing only columns of \widehat{D} . Further, $\text{card}(D)$ denotes the number of columns of D .
10. The least common multiple of a set of natural numbers is the smallest nonzero natural number that is a multiple of all the elements in the set.

3 Minimization Problem

We want to solve the linearly constrained problem

$$\min_{x \in \mathbf{X}} f(x) \quad (1a)$$

$$\mathbf{X} \triangleq \{x \in \mathbb{R}^n \mid l \leq Qx \leq u; l, u \in \mathbb{R}^{n_c} \cup \{\pm\infty\}; l < u; Q \in \mathbb{Q}^{n_c \times n}\} \quad (1b)$$

where the cost function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is (at least) Lipschitz continuous and the number of constraints n_c is finite.

We assume that the function $f(\cdot)$ cannot be evaluated exactly, but that it can be approximated by functions $f_N: \mathbb{R}^n \rightarrow \mathbb{R}$, where $N \in \mathbb{N}^q$ is an integer vector of fixed dimension $q \in \mathbb{N}$ that contains the number of iterations of the PDE, ODE, and algebraic equation solvers. We will assume that $f(\cdot)$ and its approximating functions $\{f_N(\cdot)\}_{N \in \mathbb{N}^q}$ have the following properties.

Assumption 3.1

1. There exists an error bound function $\varphi: \mathbb{N}^q \rightarrow \mathbb{R}_+$ such that for any bounded set $\mathbf{S} \subset \mathbf{X}$, there exists an $N_{\mathbf{S}} \in \mathbb{N}^q$ and a scalar $K_{\mathbf{S}} \in (0, \infty)$ such that for all $x \in \mathbf{S}$ and for all $N \in \mathbb{N}^q$, with $N \geq N_{\mathbf{S}}$ ⁴,

$$|f_N(x) - f(x)| \leq K_{\mathbf{S}} \varphi(N). \quad (2)$$

Furthermore,

$$\lim_{N \rightarrow \infty} \varphi(N) = 0. \quad (3)$$

2. The function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is at least locally Lipschitz continuous. □

Remark 3.2 The functions $f_N: \mathbb{R}^n \rightarrow \mathbb{R}$ may be discontinuous. □

In the Appendix, we give a few examples of how the error bound function arises in specific optimization problems.

Next, we state an assumption on the level sets of the family of approximate functions. To do so, we first define the notion of a level set.

Definition 3.3 (Level Set) Given a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and an $\alpha \in \mathbb{R}$, such that $\alpha \geq \inf_{x \in \mathbb{R}^n} f(x)$, we will say that the set $\mathbf{L}_{\alpha}(f) \subset \mathbb{R}^n$, defined as

$$\mathbf{L}_{\alpha}(f) \triangleq \{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}, \quad (4)$$

is a level set of $f(\cdot)$, parametrized by α . □

⁴For $N \in \mathbb{N}^q$, by $N \geq N_{\mathbf{S}}$, we mean that $N^i \geq N_{\mathbf{S}}^i$, for all $i = 1, \dots, q$.

Assumption 3.4 (Compactness of Level Sets) *Let $\{f_N(\cdot)\}_{N \in \mathbb{N}^q}$ be as in Assumption 3.1 and let $\mathbf{X} \subset \mathbb{R}^n$ be the constraint set. Let $x_0 \in \mathbf{X}$ be the initial iterate and $N_0 \in \mathbb{N}^q$ be the initial number of solver iterations. Then, we assume that there exists a compact set $\mathbf{C} \subset \mathbb{R}^n$ such that*

$$\mathbf{L}_{f_{N_0}(x_0)}(f_N) \cap \mathbf{X} \subset \mathbf{C}, \quad \forall N \geq N_0. \quad (5)$$

□

4 Generalized Pattern Search Algorithms

4.1 Geometric Aspects of the Algorithms

A major aspect of any GPS algorithm is the rule for generating the meshes on which the searches are conducted. The main difference between our rule for mesh generation and those of others, such as the one of Audet's and Dennis [1], is that we use a different rule for mesh refinement, which results in our meshes being nested, and hence simplifies the explanation of the geometry of mesh generation. As far as we can tell, our simplification has no impact on computational efficiency.

The k -th iteration of our GPS algorithms has the following structure. We begin with the current iterate x_k , with the number of iterations for the PDE, ODE, and algebraic equation solvers N , and with the mesh \mathbb{M}_k . A set-valued map is used to select a finite subset of mesh points in \mathbb{M}_k , for the so-called “global search”. If this set contains a point x' such that $f_N(x') < f_N(x_k)$, then we set $x_{k+1} = x'$, $\mathbb{M}_{k+1} = \mathbb{M}_k$, and update the index k to $k + 1$. If the global search set does not yield a point of lower cost, we proceed to a “local search”, which consists of evaluating $f_N(\cdot)$ on a set of neighbors of x_k in the mesh \mathbb{M}_k . If a point x' of lower cost is found, then we set $x_{k+1} = x'$, $\mathbb{M}_{k+1} = \mathbb{M}_k$, and update the index k to $k + 1$. If the local search also fails to produce an improvement, then the mesh \mathbb{M}_k is subdivided to yield a finer mesh \mathbb{M}_{k+1} , and the number of iterations of the PDE, ODE, and algebraic equation solvers, N , are increased according to a prescribed rule. After updating k to $k + 1$, the entire process is repeated.

We will now flesh out the geometric details of our GPS algorithms. We begin with the construction of the meshes.

4.1.1 Generation of the Meshes

Before we can explain how the mesh is to be generated, we must introduce the notions of a positive combination and of a positive span, as defined by Davis [5], and that of a base direction matrix.

Definition 4.1 (Positive Combination, Positive Span)

1. A positive combination of vectors $\{v_i\}_{i=1}^p$ is a linear combination $\sum_{i=1}^p \lambda_i v_i$ with $\lambda_i \geq 0$ for all $i \in \{1, \dots, p\}$.

2. A positive span for a subspace $\mathbf{S} \subset \mathbb{R}^n$ is a set of vectors $\{v_i\}_{i=1}^p$ such that every $x \in \mathbf{S}$ can be expressed as a positive combination of the vectors $\{v_i\}_{i=1}^p$. The matrix defined by $V \triangleq [v_1, \dots, v_p]$ is said to be a positive spanning matrix.
3. Let the subspace $\mathbf{S} \subset \mathbb{R}^n$ be of dimension m and $V \in \mathbb{R}^{n \times p}$ be a positive spanning matrix for \mathbf{S} . If $p = m + 1$, then V is said to be a minimal positive spanning matrix.

□

In [5, 4], a positive basis for a subspace $\mathbf{S} \subset \mathbb{R}^n$ is defined as a set of positively independent vectors whose positive span is \mathbf{S} . Note that a positive basis is different from a minimal positive spanning set. For example, if $\mathbf{S} = \mathbb{R}^2$, the set $\{e_1, e_2, -e_1, -e_2\}$ is a positive basis but not a minimal positive spanning set. A minimal positive spanning set is $\{e_1, e_2, -(e_1 + e_2)\}$.

We will denote by \mathbb{S} the set of all matrices whose columns positively span \mathbb{R}^n .

Next, we define a *base direction matrix*. We will use the columns of the base direction matrix to specify the mesh and hence the search directions. The base direction matrix will be fixed for all iterations.

Definition 4.2 (Base Direction Matrix) Let \mathbb{S} be the set of all matrices whose columns positively span \mathbb{R}^n . Then, the base direction matrix \hat{D} is any matrix satisfying

$$\hat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S} \quad (6)$$

where $p > n$ is any arbitrary but finite natural number.

□

Remark 4.3 The fact that the matrix \hat{D} has only rational elements makes it very easy to establish the minimal distance between distinct mesh points (Lemma 5.1). At the same time, from a computational point of view, requiring $\hat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S}$ rather than $\hat{D} \in \mathbb{R}^{n \times p} \cap \mathbb{S}$ does not result in any practical inconvenience.

□

Note that the base direction matrix \hat{D} may not be a minimal positive spanning matrix, e.g., for the one-dimensional case, $\hat{D} = [-1, 1, 1.1]$ would not be minimal. Hence, \hat{D} can be used to generate a set $\mathbb{D}_{\hat{D}}$, which we define as the set of all submatrices of \hat{D} (constructed by deleting columns of \hat{D}) whose column vectors positively span \mathbb{R}^n .

The meshes, over which our algorithms search, are defined iteratively, as follows.

Definition 4.4 (k -th Mesh) Let $x_0 \in \mathbf{X}$, $r, s_0, k \in \mathbb{N}$, with $r > 1$, $\{t_i\}_{i=0}^{k-1} \subset \mathbb{N}$, and the base direction matrix $\hat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S}$ be given, and let

$$\Delta_k \triangleq \frac{1}{r^{s_k}}, \quad (7)$$

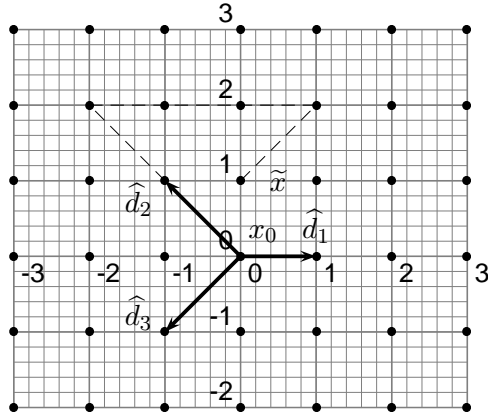


Figure 1: Minimal positive spanning matrix $\hat{D} = [\hat{d}_1, \hat{d}_2, \hat{d}_3]$ and generated mesh in \mathbb{R}^2

where for $k > 0$

$$s_k = s_0 + \sum_{i=0}^{k-1} t_i. \quad (8)$$

Then we define the mesh \mathbb{M}_k by

$$\mathbb{M}_k \triangleq \{x_0 + \Delta_k \hat{D} m \mid m \in \mathbb{N}^p\}. \quad (9)$$

□

It should be clear from the definition of the meshes that whenever $t_k > 0$, the mesh \mathbb{M}_{k+1} is obtained from the mesh \mathbb{M}_k by dividing the intervals between neighboring points of the mesh \mathbb{M}_k into r^{t_k} subintervals by adding additional mesh points. Therefore, it is clear that the meshes are nested, i.e., $\mathbb{M}_k \subset \mathbb{M}_{k+1}$ with equality if $\Delta_{k+1} = \Delta_k$.

We now present two examples: first a simple example of a mesh that is generated by a minimal positive spanning matrix, and then an example of a mesh generation using a more complicated base direction matrix \hat{D} .

Example 4.5 In Fig. 1, the base direction matrix \hat{D} is a minimal positive spanning matrix, defined by

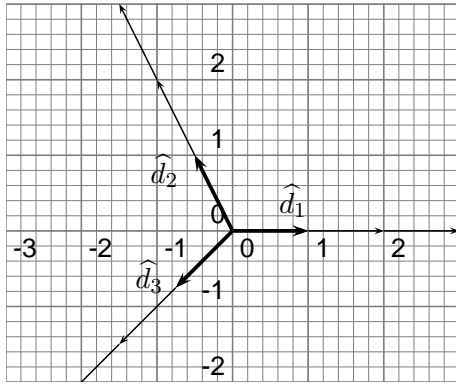
$$\hat{D} = \begin{pmatrix} \hat{d}_1 & \hat{d}_2 & \hat{d}_3 \end{pmatrix} \triangleq \begin{pmatrix} 1 & -1 & -1 \\ 0 & 1 & -1 \end{pmatrix}. \quad (10)$$

In Fig. 1, the bullets (•) are the mesh points of the mesh $\mathbb{M}_k = \{0 + 1 \hat{D} m \mid m \in \mathbb{N}^3\}$. For example, in Fig. 1, $\tilde{x} = \hat{D} m$, with $m = (3, 2, 1)^T$. □

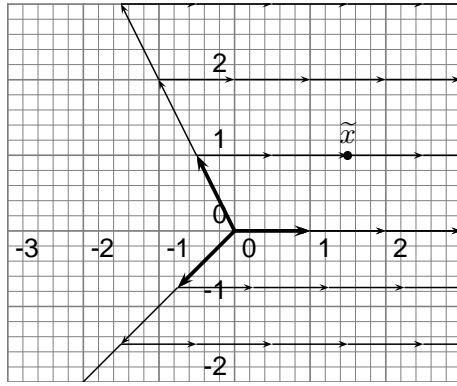
Next we present a mesh constructed using a more complicated base direction matrix.

Example 4.6 Fig. 2 shows a mesh generated using $x_0 = 0$, $\Delta_k = 1$ and the base direction matrix

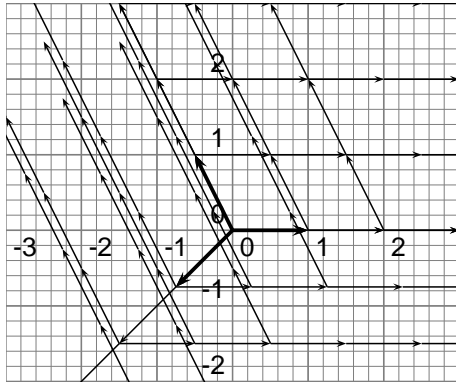
$$\hat{D} = \begin{pmatrix} \hat{d}_1 & \hat{d}_2 & \hat{d}_3 \end{pmatrix} = \begin{pmatrix} 1 & -0.5 & -0.75 \\ 0 & 1 & -0.75 \end{pmatrix}. \quad (11)$$



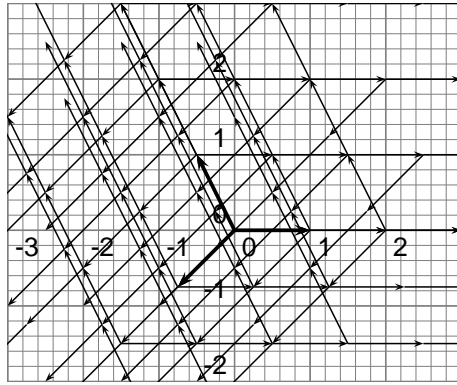
(a)



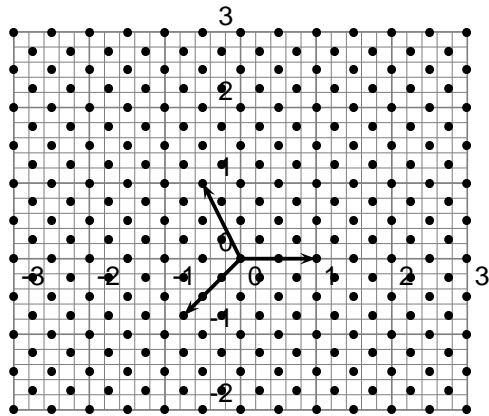
(b)



(c)



(d)



(e)

Figure 2: Generation of a mesh in \mathbb{R}^2

Fig. 2(a) shows the vectors $\{\hat{d}_i\}_{i=1}^3$ (bold arrows) and all possible mesh points of the form $\hat{D}v$ with $v = (n, 0, 0)^T$, $v = (0, n, 0)^T$, and $v = (0, 0, n)^T$ where $n \in \mathbb{N}$. Each arrow points to a mesh point and indicates how the base vectors $\{\hat{d}_i\}_{i=1}^3$ are added to obtain the mesh points. Fig. 2(b) shows the set of all mesh points of the form $\hat{D}v$ with $v = (n, m, 0)^T$ and $v = (n, 0, m)^T$ where $n, m \in \mathbb{N}$. For example, the point labeled with \tilde{x} is given by $\tilde{x} = \hat{D}v$ where $v = (2, 1, 0)^T$. In Fig. 2(c), more mesh points are drawn by adding some positive multiple of \hat{d}_2 to some mesh points that have been generated in Fig. 2(b). For clarity, not all possible mesh points are drawn. In Fig. 2(d), additional mesh points are generated by adding some positive multiple of \hat{d}_3 to some mesh points of Fig. 2(c). Fig. 2(e) finally contains all possible mesh points, now indicated by bullets (\bullet). For clarity, only the vectors $\{\hat{d}_i\}_{i=1}^3$ are drawn in Fig. 2(e). \square

4.1.2 Global and Local Search Set

We will now characterize the set-valued maps that determine the mesh points for the “global” and “local” searches. Note that the images of these maps may depend on the entire history of the computation.

Definition 4.7 (Search Direction Matrices) *Let \mathbb{S} be the set of all matrices whose column vectors positively span \mathbb{R}^n . Given a base direction matrix \hat{D} , we define the set of search direction matrices to be*

$$\mathbb{D}_{\hat{D}} \triangleq \{D \mid D \subset \hat{D} \cap \mathbb{S}\} \quad (12)$$

where the matrix D is constructed by deleting columns of \hat{D} . \square

Definition 4.8 *Let $\underline{\mathbf{X}}_k \subset \mathbb{R}^n$ and $\underline{\Delta}_k \subset \mathbb{Q}_+$ be the sets of all sequences containing $k+1$ elements, let \mathbb{M}_k be the current mesh, let $\mathbb{D}_{\hat{D}}$ be the set of search direction matrices, and let $N \in \mathbb{N}^q$ be the number of solver iterations.*

1. We define the global search set map to be any set-valued map

$$\gamma_k : \underline{\mathbf{X}}_k \times \underline{\Delta}_k \times \mathbb{N}^q \rightarrow (2^{\mathbb{M}_k} \cap \mathbf{X}) \cup \emptyset \quad (13a)$$

whose image $\gamma_k(\underline{x}_k, \underline{\Delta}_k, N)$ contains only a finite number of mesh points.

2. We define the local search direction map to be any map

$$\delta_{\hat{D},k} : \underline{\mathbf{X}}_k \times \underline{\Delta}_k \rightarrow \mathbb{D}_{\hat{D}}. \quad (13b)$$

3. We will call $\mathcal{G}_k \triangleq \gamma_k(\underline{x}_k, \underline{\Delta}_k, N)$ the global search set.

4. With $D_k = \delta_{\hat{D},k}(\underline{x}_k, \underline{\Delta}_k)$, we will call

$$\mathcal{L}_k \triangleq \{x_k + \Delta_k D_k e_j \mid j = 1, \dots, \text{card}(D_k)\} \cap \mathbf{X} \quad (13c)$$

the local search set. \square

Remark 4.9

1. The map $\gamma_k(\cdot, \cdot, \cdot)$ can be dynamic in the sense that if $\{x_{k_i}\}_{i=0}^I \triangleq \gamma_k(\underline{x}_k, \underline{\Delta}_k, N)$, then the rule for selecting $x_{k_{\hat{i}}}$, $1 \leq \hat{i} \leq I$, can depend on $\{x_{k_i}\}_{i=0}^{\hat{i}-1}$ and $\{f_N(x_{k_i})\}_{i=0}^{\hat{i}-1}$. It is only important that the global search terminates after a finite number of computations, and that $\mathcal{G}_k \subset (2^{\mathbb{M}_k} \cap \mathbf{X}) \cup \emptyset$.
2. As we shall see, the global search affects only the efficiency of the algorithm but not its convergence properties. Any heuristic procedure that leads to a finite number of function evaluations can be used for $\gamma_k(\cdot, \cdot, \cdot)$.
3. The empty set is included in the range of $\gamma_k(\cdot, \cdot, \cdot)$ to allow omitting the global search.
4. Since the range of $\delta_{\hat{D},k}(\cdot, \cdot)$ is $\mathbb{D}_{\hat{D}}$, any image of $\delta_{\hat{D},k}(\cdot, \cdot)$ is a positive spanning matrix.

□

4.2 A Model Adaptive Precision GPS Algorithm

We are now ready to present a model generalized pattern search algorithm with adaptive precision function evaluations.

Algorithm 4.10 (Model GPS Algorithm)

-
- Data:** Initial iterate $x_0 \in \mathbf{X}$;
 Mesh size divider $r \in \mathbb{N}$, with $r > 1$;
 Initial mesh size exponent $s_0 \in \mathbb{N}$;
 Base direction matrix $\hat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S}$ (see Definition 4.2).
- Maps:** Global search set map $\gamma_k: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \times \mathbb{N}^q \rightarrow (2^{\mathbb{M}_k} \cap \mathbf{X}) \cup \emptyset$;
 Local search direction map $\delta_{\hat{D},k}: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \rightarrow \mathbb{D}_{\hat{D}}$ (see Definition 4.8).
 Function $\rho: \mathbb{R}_+ \rightarrow \mathbb{N}^q$ (to assign N), such that the composition $\varphi \circ \rho: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is strictly monotone decreasing and satisfies $\varphi(\rho(\Delta))/\Delta \rightarrow 0$, as $\Delta \rightarrow 0$.
- Step 0:** Initialize $k = 0$, $\Delta_0 = 1/r^{s_0}$, and $N = \rho(1)$.
- Step 1:** Global Search
 Construct the global search set $\mathcal{G}_k = \gamma_k(\underline{x}_k, \underline{\Delta}_k, N)$.
 If $f_N(x') < f_N(x_k)$ for any $x' \in \mathcal{G}_k$, go to Step 3; else, go to Step 2.
- Step 2:** Local Search
 Construct the search direction matrix $D_k = \delta_{\hat{D},k}(\underline{x}_k, \underline{\Delta}_k)$.
 Construct $\mathcal{L}_k \triangleq \{x_k + \Delta_k D_k e_j \mid j = 1, \dots, \text{card}(D_k)\} \cap \mathbf{X}$ and evaluate $f_N(\cdot)$ for any $x' \in \mathcal{L}_k$ until some $x' \in \mathcal{L}_k$ satisfying $f_N(x') < f_N(x_k)$ is obtained, or until all points in \mathcal{L}_k are evaluated.
- Step 3:** Parameter Update
 If there exists an $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ satisfying $f_N(x') < f_N(x_k)$, set $x_{k+1} = x'$, $s_{k+1} = s_k$, $\Delta_{k+1} = \Delta_k$, and do not change N ; else, set $x_{k+1} = x_k$, $s_{k+1} = s_k + t_k$, with $t_k \in \mathbb{N}_+$ arbitrary, $\Delta_{k+1} = 1/r^{s_{k+1}}$, $N = \rho(\Delta_{k+1}/\Delta_0)$.
- Step 4:** Replace k by $k + 1$, and go to Step 1.
-

Remark 4.11

1. If the optimization is started with $N = \rho(1)$ too large, the computation time may become unnecessary large. Therefore, in implementing the Model GPS Algorithm, one may allow to redefine the function $\rho(\cdot)$ by $\rho(\cdot) \leftarrow c\rho(\cdot)$, with $c \in (0, 1)$, to decrease the initial number of solver iterations. Redefining the function $\rho(\cdot)$ is allowed over a preset number of GPS iterations.
2. To ensure that N does not depend on the scaling of Δ_0 , we normalized the argument of $\rho(\cdot)$. In particular, we want to decouple the number of iterations of the solvers from the user's choice of the initial mesh divider.
3. Audet and Dennis [1] increase and decrease the mesh divider using the formula $\Delta_{k+1} = \tau^m \Delta_k$ where $\tau \in \mathbb{Q}$, $\tau > 1$, and m is any element of \mathbb{Z} . Thus, our mesh construction is a special case of Audet's and Dennis' construction since we set $\tau = 1/r$, with $r \in \mathbb{N}_+$, $r \geq 2$ (so that, $\tau < 1$) and $m \in \mathbb{N}$. We prefer our construction because it leads to a simpler geometric explanation. In the Appendix, we present a modified version of the algorithm of Audet and Dennis, and show that our analysis remains valid.
4. In Step 2, once a decrease of the cost function is obtained, one can proceed to Step 3. However, one is allowed to evaluate the approximating cost function at more points in \mathcal{L}_k in an attempt to obtain a bigger reduction in cost. However, one is allowed to proceed to Step 3 only after either a cost decrease has been found, or after *all* points in \mathcal{L}_k are tested.
5. In Step 3, we are not restricted to accepting the $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ that gives lowest cost value. But the mesh divider Δ_k is reduced *only* if there exists no $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ satisfying $f_N(x') < f_N(x_k)$. □

4.3 An Extension of the Hooke-Jeeves Algorithm

To illustrate the use of our Model GPS Algorithm 4.10, we will now use it to obtain an extension of the Hooke-Jeeves algorithm [8]. To simplify exposition, we will assume that $\mathbf{X} = \mathbb{R}^n$.

4.3.1 Algorithm Parameters \hat{D} , r , s_0 and t_k

Hooke and Jeeves decrease the “current step size” ($\Delta \in \mathbb{R}_+$ in [8]) by a factor $\rho \in (0, 1)$, when necessary. To fit their algorithm into our framework, we have to set $\rho \triangleq 1/q$ for some $q \in \mathbb{N}_+ \setminus \{1\}$ ⁵ and restrict the initial value of their variable Δ to take on rational values only⁶.

⁵The restriction $\rho \triangleq 1/q$ is not serious because one usually has no knowledge that justifies requiring another value.

⁶In numerical computer programs, the restriction $\Delta \in \mathbb{Q}_+$ is automatically fulfilled since irrational numbers cannot be represented.

In view of the above, for our extension of the Hooke-Jeeves algorithm, we define our base direction matrix as $\hat{D} \triangleq \Delta [+e_1, -e_1, \dots, +e_n, -e_n]$ (where Δ is the *initial* value of the “step size” in [8]) and our other parameters to be $r \triangleq q$, $s_0 = 0$, and $t_k \in \{0, 1\}$, for all $k \in \mathbb{N}$.

4.3.2 Map for Exploratory Moves

To facilitate the algorithm explanation, we first introduce a set-valued map $E: \mathbb{R}^n \times \mathbb{Q}_+ \times \mathbb{N}^q \rightarrow 2^{\mathbb{M}_k}$, which defines the “exploratory moves” in [8]. The map $E: \mathbb{R}^n \times \mathbb{Q}_+ \times \mathbb{N}^q \rightarrow 2^{\mathbb{M}_k}$ will then be used in Section 4.3.3 to define the global search set map and, under conditions to be seen in Section 4.3.4, the local search direction map as well.

Algorithm 4.12 (Map $E: \mathbb{R}^n \times \mathbb{Q}_+ \times \mathbb{N}^q \rightarrow 2^{\mathbb{M}_k}$ for “Exploratory Moves”)

Parameter: Base direction matrix $\hat{D} = \Delta [+e_1, -e_1, \dots, +e_n, -e_n] \in \mathbb{Q}^{n \times 2n}$
 (Δ being the initial step size of [8]).

Input: Base point $x \in \mathbb{R}^n$;
 Mesh divider $\Delta_k \in \mathbb{Q}_+$;

Output: Set of trial points \mathcal{T} .

Step 0: Initialize $\mathcal{T} = \emptyset$.

Step 1: For $i = 1, \dots, n$
 Set $\tilde{x} = x + \Delta_k \hat{D} e_{2i-1}$ and $\mathcal{T} \leftarrow \mathcal{T} \cup \{\tilde{x}\}$.
 If $f_N(\tilde{x}) < f_N(x)$
 Set $x = \tilde{x}$.
 else
 Set $\tilde{x} = x + \Delta_k \hat{D} e_{2i}$ and $\mathcal{T} \leftarrow \mathcal{T} \cup \{\tilde{x}\}$.
 If $f_N(\tilde{x}) < f_N(x)$
 Set $x = \tilde{x}$.
 end if.
 end if.
 end for.

Step 2: Return \mathcal{T} .

Thus, $E(x, \Delta_k, N) = \mathcal{T}$.

4.3.3 Global Search Set Map $\gamma_k: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \times \mathbb{N}^q \rightarrow 2^{\mathbb{M}_k}$

The global search set map $\gamma_k(\cdot, \cdot, \cdot)$ is defined as below. Because $\gamma_0(\cdot, \cdot, \cdot)$ depends on x_{-1} , we need to introduce x_{-1} , which we define as $x_{-1} \triangleq x_0$.

Algorithm 4.13 (Global Search Set Map $\gamma_k: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \times \mathbb{N}^q \rightarrow 2^{\mathbb{M}_k}$)

Map: Map for “exploratory moves” $E: \mathbb{R}^n \times \mathbb{Q}_+ \times \mathbb{N}^q \rightarrow 2^{\mathbb{M}_k}$.
Input: Previous and current iterate, $x_{k-1} \in \mathbb{R}^n$ and $x_k \in \mathbb{R}^n$;
 Mesh divider $\Delta_k \in \mathbb{Q}_+$;
 Number of solver iterations $N \in \mathbb{N}^q$.
Output: Global search set \mathcal{G}_k .
Step 1: Set $x = x_k + (x_k - x_{k-1})$.
Step 2: Compute $\mathcal{G}_k = E(x, \Delta_k, N)$.
Step 3: If $(\min_{x \in \mathcal{G}_k} f_N(x)) > f_N(x_k)$
 Set $\mathcal{G}_k \leftarrow \mathcal{G}_k \cup E(x_k, \Delta_k, N)$.
 end if.
Step 4: Return \mathcal{G}_k .

Thus, $\gamma_k(\underline{x}_k, \underline{\Delta}_k, N) = \mathcal{G}_k$.

4.3.4 Local Search Direction Map $\delta_{\hat{D},k}: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \rightarrow \mathbb{D}_{\hat{D}}$

If the global search, as defined by Algorithm 4.13, has failed in reducing $f_N(\cdot)$, then Algorithm 4.13 has constructed a set \mathcal{G}_k that contains the set $\{x_k + \Delta_k \hat{D} e_i \mid i = 1, \dots, 2n\}$. This is because in the evaluation of $E(x_k, \Delta_k, N)$, all “if(·)” statements yield **false**, and, hence, one has constructed $\{x_k + \Delta_k \hat{D} e_i \mid i = 1, \dots, 2n\} = E(x_k, \Delta_k, N)$.

Because the columns of \hat{D} span \mathbb{R}^n positively, it follows that the search on the set $\{x_k + \Delta_k \hat{D} e_i \mid i = 1, \dots, 2n\}$ is a local search. Hence, the constructed set

$$\mathcal{L}_k \triangleq \{x_k + \Delta_k \hat{D} e_i \mid i = 1, \dots, 2n\} \subset \mathcal{G}_k \quad (14)$$

is a local search set. Consequently, $f_N(\cdot)$ has already been evaluated at all points of \mathcal{L}_k (during the construction of \mathcal{G}_k) and, hence, one does not need to evaluate $f_N(\cdot)$ again in a local search. In view of (13c) and (14), the local search direction map is given by $D_k = \delta_{\hat{D},k}(\underline{x}_k, \underline{\Delta}_k) \triangleq \hat{D}$.

4.3.5 Parameter Update

The point x' in Step 3 of the GPS Model Algorithm 4.10 corresponds to $x' \triangleq \arg \min_{x \in \mathcal{G}_k} f_N(x)$ in the Hooke-Jeeves algorithm. (Note that $\mathcal{L}_k \subset \mathcal{G}_k$ if a local search has been done as explained in the above paragraph.)

5 Convergence Results

5.1 Unconstrained Minimization

We will now establish the convergence properties of the Model GPS Algorithm 4.10 on unconstrained minimization problems, i.e., for $\mathbf{X} = \mathbb{R}^n$.

First, we will show that for any mesh \mathbb{M}_k , the minimal Euclidean distance between all distinct mesh points is bounded from below by a constant times the mesh divider Δ_k .

Lemma 5.1 (Minimal Distance between Distinct Mesh Points) *Consider the sequences $\{\Delta_k\}_{k=0}^\infty \subset \mathbb{Q}_+$ of mesh dividers, and $\{\mathbb{M}_k\}_{k=0}^\infty$ of meshes. Then there exists a constant $c > 0$, independent of k , such that*

$$\min_{\substack{u \neq v \\ u, v \in \mathbb{M}_k}} \|u - v\| \geq \Delta_k c. \quad (15)$$

Proof. By Definition 4.4, for any given k , we have $\mathbb{M}_k \triangleq \{x_0 + \Delta_k \hat{D} m \mid m \in \mathbb{N}^p\}$ where $\hat{D} \in \mathbb{Q}^{n \times p}$ is fixed for all k . Let l be the least common multiple of all denominators of the elements of \hat{D} . Then, $\hat{Z} \triangleq l \hat{D}$ is in $\mathbb{Z}^{n \times p}$. Furthermore, any pair of mesh points u, v can be represented as $u \triangleq x_0 + \Delta_k \hat{D} m_u$ and $v \triangleq x_0 + \Delta_k \hat{D} m_v$, where $m_u, m_v \in \mathbb{N}^p$. Hence,

$$\begin{aligned} \min_{\substack{u \neq v \\ u, v \in \mathbb{M}_k}} \|u - v\| &= \min_{\substack{\|\hat{D}(m_u - m_v)\| \neq 0 \\ m_u, m_v \in \mathbb{N}^p}} \Delta_k \|\hat{D}(m_u - m_v)\| \\ &= \min_{\substack{\|\hat{D}m\| \neq 0 \\ m \in \mathbb{Z}^p}} \Delta_k \|\hat{D}m\| = \frac{\Delta_k}{l} \min_{\substack{\|\hat{Z}m\| \neq 0 \\ m \in \mathbb{Z}^p}} \|\hat{Z}m\| \geq \frac{\Delta_k}{l}. \end{aligned} \quad (16)$$

The inequality holds because $\hat{Z}m$ is a nonzero integer vector. \square

The following corollary follows directly from Lemma 5.1 and will be used to show that $\Delta_k \rightarrow 0$ as $k \rightarrow \infty$.

Corollary 5.2 *Any bounded subset of a mesh \mathbb{M}_k contains only a finite number of mesh points.* \square

Proposition 5.3 *Consider the sequence of mesh dividers $\{\Delta_k\}_{k=0}^\infty \subset \mathbb{Q}_+$ constructed by Model GPS Algorithm 4.10. Then, the mesh dividers satisfy $\liminf_{k \rightarrow \infty} \Delta_k = 0$.*

Proof. By (7), $\Delta_k = 1/r^{s_k}$, where $r \in \mathbb{N}$ with $r > 1$, and $\underline{s}_k \subset \mathbb{N}$ is a nondecreasing sequence. For the sake of contradiction, suppose that there exists a $\Delta_{k^*} \in \mathbb{Q}_+$, such that $\Delta_k \geq \Delta_{k^*}$ for all $k \in \mathbb{N}$. Then there exists a corresponding $s_{k^*} = \max_{k \in \mathbb{N}} s_k$, and the finest possible mesh is $\mathbb{M}_{k^*} \triangleq \{x_0 + (1/r^{s_{k^*}}) \hat{D} m \mid m \in \mathbb{N}^p\}$.

Next, since by Assumption 3.4, there exists a compact set \mathbf{C} , such that $\mathbf{L}_{f_{N_0}(x_0)}(f_N) \cap \mathbf{X} \subset \mathbf{C}$ for all $N \geq N_0 = \rho(1)$, it follows from Corollary 5.2 that $\mathbb{M}_{k^*} \cap \mathbf{L}_{f_{N_0}(x_0)}(f_N)$ contains only a finite number of points for any $N \geq \rho(1)$. Thus, at least one point in \mathbb{M}_{k^*} must belong to the sequence $\{x_k\}_{k=0}^\infty$ infinitely many times. Furthermore, because $\{s_k\}_{k=0}^\infty \subset \mathbb{N}$ is nondecreasing with s_{k^*} being its maximal element, it follows that $N = N^* = \rho(\Delta_{k^*}/\Delta_0)$ for all iterations $k \geq k^*$. Hence the sequence $\{f_{N^*}(x_k)\}_{k=0}^\infty$ cannot be strictly monotone

decreasing, which contradicts the constructions in Algorithm 4.10. \square

Having shown that $\liminf_{k \rightarrow \infty} \Delta_k = 0$, we can introduce the notion of a refining subsequence as used by Audet and Dennis [1].

Definition 5.4 (Refining Subsequence) *Consider a sequence $\{x_k\}_{k=0}^\infty$ constructed by Model GPS Algorithm 4.10. We will say that the subsequence $\{x_k\}_{k \in \mathbf{K}}$ is the refining subsequence, if $\Delta_{k+1} < \Delta_k$ for all $k \in \mathbf{K}$, and $\Delta_{k+1} = \Delta_k$ for all $k \notin \mathbf{K}$. \square*

When the cost function $f(\cdot)$ is only locally Lipschitz continuous, we, as well as Audet and Dennis [1], only get a weak characterization of limit points of refining sequences. As we will now see.

We recall the definition of Clarke's generalized directional derivative [3]:

Definition 5.5 (Clarke's Generalized Directional Derivative) *Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be locally Lipschitz continuous at the point $x^* \in \mathbb{R}^n$. Then, Clarke's generalized directional derivative of $f(\cdot)$ at x^* in the direction $h \in \mathbb{R}^n$ is defined by*

$$d^0 f(x^*; h) \triangleq \limsup_{\substack{x \rightarrow x^* \\ t \downarrow 0}} \frac{f(x + t h) - f(x)}{t}. \quad (17)$$

\square

Theorem 5.6 *Suppose that Assumptions 3.1 and 3.4 are satisfied and let $x^* \in \mathbb{R}^n$ be an accumulation point of a refining subsequence $\{x_k\}_{k \in \mathbf{K}}$, constructed by Model GPS Algorithm 4.10. Let d be any column of the base direction matrix \widehat{D} along which $f_N(\cdot)$ was evaluated for infinitely many iterates in the subsequence $\{x_k\}_{k \in \mathbf{K}}$. Then,*

$$d^0 f(x^*; d) \geq 0. \quad (18)$$

Proof. Let $\{x_k\}_{k \in \mathbf{K}}$ be the refining subsequence and, WLOG, suppose that $x_k \xrightarrow{\mathbf{K}} x^*$. By Assumption 3.4, there exists a compact set \mathbf{C} such that $\mathbf{L}_{f_{N_0}(x_0)}(f_N) \cap \mathbf{X} \subset \mathbf{C}$ for all $N \geq N_0 = \rho(1)$. Therefore, by Assumption 3.1, there exists an $N_{\mathbf{L}} \in \mathbb{N}^q$ and a scalar $K_{\mathbf{L}} \in (0, \infty)$ such that, for all $x \in \mathbf{C}$ and for all $N \geq N_{\mathbf{L}}$, we have $|f_N(x) - f(x)| \leq K_{\mathbf{L}} \varphi(N)$. Because $f(\cdot)$ is locally Lipschitz continuous, its directional derivative $d^0 f(\cdot; \cdot)$

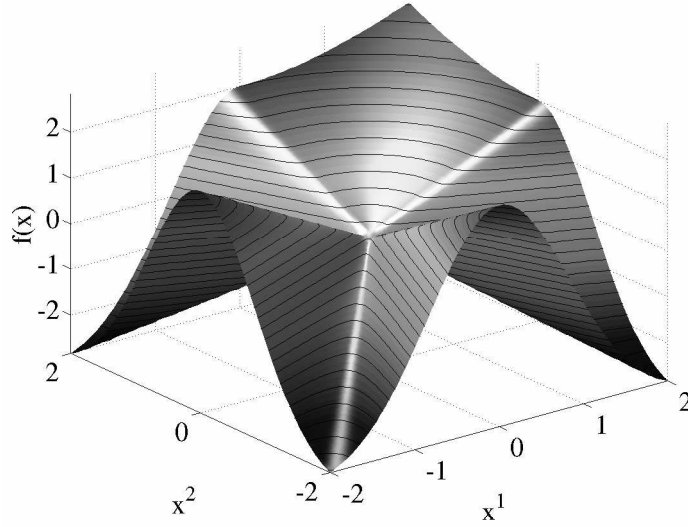


Figure 3: Visualization of equation (20)

exists. Hence, noting that $N = N_k = \rho(\Delta_k/\Delta_0)$,

$$\begin{aligned}
 d^0 f(x^*; d) &\triangleq \limsup_{\substack{x \rightarrow x^* \\ t \downarrow 0}} \frac{f(x + t d) - f(x)}{t} \\
 &\geq \limsup_{k \in \mathbf{K}} \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k} \\
 &\geq \limsup_{k \in \mathbf{K}} \frac{f_N(x_k + \Delta_k d) - f_N(x_k) - 2 K_{\mathbf{L}} \varphi(N)}{\Delta_k} \\
 &\geq \limsup_{k \in \mathbf{K}} \frac{f_N(x_k + \Delta_k d) - f_N(x_k)}{\Delta_k} - \limsup_{k \in \mathbf{K}} 2 K_{\mathbf{L}} \frac{\varphi(N)}{\Delta_k} \\
 &\geq - \limsup_{k \in \mathbf{K}} 2 K_{\mathbf{L}} \frac{\varphi(N)}{\Delta_k}.
 \end{aligned} \tag{19}$$

The last inequality holds because $\{x_k\}_{k \in \mathbf{K}}$ is a refining subsequence. Since by Proposition 5.3, $\Delta_k \rightarrow 0$, it follows from the constructions in Model GPS Algorithm 4.10 that $\varphi(N)/\Delta_k \rightarrow_{\mathbf{K}} 0$. \square

Remark 5.7 Note that (18) is not a standard optimality condition since it holds only for certain directions d . Consider, for example, the Lipschitz continuous function

$$f(x) \triangleq \begin{cases} \|x\|, & \text{if } x^1 > 0 \text{ and } x^2 > 0, \\ \|x\| \cos(4 \arccos(x^1/\|x\|)), & \text{otherwise,} \end{cases} \tag{20}$$

which is shown in Fig. 3. This function is not differentiable at the origin, but it does have directional derivatives everywhere. At the origin $x^* = 0$, we have $d f(x^*; d) = 1$ for

$d \in \{\pm e_1, \pm e_2\}$, but the directional derivative along $s = (-1, -1)^T$ is $df(x^*; s) = -\sqrt{2}$. Using the Hooke-Jeeves algorithm with initial value $x_0 = (-1, 0)^T$ and $\Delta = \Delta_0 = 1$, we would converge to the origin, a point that possess some negative directional derivatives. \square

We now state that pattern search algorithms with adaptive precision function evaluations converge to stationary points.

Theorem 5.8 (Convergence to a Stationary Point) *Suppose that Assumptions 3.1 and 3.4 are satisfied and, in addition, that $f(\cdot)$ is once continuously differentiable. Let $x^* \in \mathbb{R}^n$ be an accumulation point of a refining subsequence $\{x_k\}_{k \in \mathbf{K}}$, constructed by Model GPS Algorithm 4.10. Then,*

$$\nabla f(x^*) = 0. \quad (21)$$

Proof. Since $f(\cdot)$ is once continuously differentiable, we have $d^0 f(x^*; h) = df(x^*; h) = \langle \nabla f(x^*), h \rangle$. Now, let $\mathbb{D}_{\widehat{D}}$ be the set of search direction matrices, and let $D^* \in \mathbb{D}_{\widehat{D}}$ be any positive spanning matrix that is used infinitely many times in conjunction with the refining subsequence $\{x_k\}_{k \in \mathbf{K}}$. Since the number of distinct columns in $\mathbb{D}_{\widehat{D}}$ is finite, there must be at least one such D^* . It follows from Theorem 5.6 that $0 \leq \langle \nabla f(x^*), d^* \rangle$ for all $d^* \in D^*$. Let l denote the number of columns of D^* . Then, because the columns of D^* positively span \mathbb{R}^n , we can express any $h \in \mathbb{R}^n$, as follows,

$$h = \sum_{i=1}^l \alpha_i d_i^*, \quad d_i^* \in D^*, \quad \alpha_i \geq 0, \quad \forall i \in \{1, \dots, l\}. \quad (22a)$$

Hence, $0 \leq \langle \nabla f(x^*), h \rangle$. Similarly, we can express the vector $-h$, as follows,

$$-h = \sum_{i=1}^l \beta_i d_i^*, \quad d_i^* \in D^*, \quad \beta_i \geq 0, \quad \forall i \in \{1, \dots, l\}. \quad (22b)$$

Hence, $0 \geq \langle \nabla f(x^*), h \rangle$, which implies $0 = \langle \nabla f(x^*), h \rangle$, and, since h is arbitrary, that $\nabla f(x^*) = 0$. \square

5.2 Linearly Constrained Minimization

We now extend our convergence proofs to the linearly constrained problem (1), by following the arguments in Audet and Dennis [1].

First, we introduce the notion of a tangent cone and a normal cone, which are defined as follows:

Definition 5.9 (Tangent and Normal Cone)

1. Let $\mathbf{X} \subset \mathbb{R}^n$. Then, we define the tangent cone to \mathbf{X} at a point $x^* \in \mathbf{X}$ by

$$\mathbf{T}_{\mathbf{X}}(x^*) \triangleq \overline{\{\mu(x - x^*) \mid \mu \geq 0, x \in \mathbf{X}\}}. \quad (23a)$$

2. Let $\mathbf{T}_{\mathbf{X}}(x^*)$ be as above. Then, we define the normal cone to \mathbf{X} at $x^* \in \mathbf{X}$ by

$$\mathbf{N}_{\mathbf{X}}(x^*) \triangleq \{v \in \mathbb{R}^n \mid \forall t \in \mathbf{T}_{\mathbf{X}}(x^*), \langle v, t \rangle \leq 0\}. \quad (23b)$$

□

Next, we introduce the concept of conformity of a pattern to a constraint set (see [1]), which will enable us to extend the convergence results for our Model GPS Algorithm 4.10 from unconstrained optimization problems to linearly constrained optimization problems.

Definition 5.10 The function $\delta_{\hat{D},k}: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \rightarrow \mathbb{D}_{\hat{D}}$ is said to conform to the feasible set \mathbf{X} , if for some $\rho > 0$ and for each $x^* \in \partial\mathbf{X}$ satisfying $\|x^* - x_k\| < \rho$, the tangent cone $\mathbf{T}_{\mathbf{X}}(x^*)$ can be generated by nonnegative linear combinations of the columns of a subset $D_{x^*}(x_k) \subset D_k = \delta_{\hat{D},k}(\underline{x}_k, \underline{\Delta}_k)$.

Furthermore, we define $D_{x^*}(\cdot)$ to be such that all its columns belong to $\mathbf{T}_{\mathbf{X}}(x^*)$. □

Remark 5.11 The definition that all columns of $D_{x^*}(\cdot)$ belong to $\mathbf{T}_{\mathbf{X}}(x^*)$ facilitates the extension of Theorem 5.6 to the constraint case. □

We can now state that the accumulation points generated by Model GPS Algorithm 4.10 are feasible stationary points of problem (1).

Theorem 5.12 (Convergence to a Feasible Stationary Point)

Suppose Assumptions 3.1 and 3.4 are satisfied and that $f(\cdot)$ is once continuously differentiable. Let $x^* \in \mathbf{X}$ be an accumulation point of a refining subsequence $\{x_k\}_{k \in \mathbf{K}}$ constructed by Model GPS Algorithm 4.10 in solving problem (1).

If there exists a $k^* \in \mathbb{N}$ such that for all $k > k^*$, the local search direction maps $\delta_{\hat{D},k}: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \rightarrow \mathbb{D}_{\hat{D}}$ conform to the feasible set \mathbf{X} , then

$$\langle \nabla f(x^*), t \rangle \geq 0, \quad \forall t \in \mathbf{T}_{\mathbf{X}}(x^*), \quad (24a)$$

and

$$-\nabla f(x^*) \in \mathbf{N}_{\mathbf{X}}(x^*). \quad (24b)$$

Proof. If x^* is in the interior of \mathbf{X} , then the result reduces to Theorem 5.8.

Let $x^* \in \partial\mathbf{X}$ and let $D_{x^*}(x_k)$ be as in Definition 5.10. Since the family of maps $\{\delta_{\hat{D},k}(\cdot, \cdot)\}_{k \in \mathbf{K}, k > k^*}$ conforms to the feasible set \mathbf{X} and since there are only finitely many linear constraints, we have that $D_{x^*}(x_k)$ converges to $D_{x^*}(x^*)$, as $x_k \xrightarrow{\mathbf{K}'} x^*$, for some

infinite subset $\mathbf{K}' \subset \mathbf{K}$. By Theorem 5.6, we have $\langle \nabla f(x^*), d \rangle \geq 0$ for all $d \in D_{x^*}(x^*)$. Furthermore, it follows from the conformity of the family of local search direction maps $\{\delta_{\hat{D},k}(\cdot, \cdot)\}_{k \in \mathbf{K}, k > k^*}$, that every $t \in \mathbf{T}_{\mathbf{X}}(x^*)$ is a nonnegative linear combination of columns of $D_{x^*}(x^*)$. Therefore, $\langle \nabla f(x^*), t \rangle \geq 0$. It follows directly that $\langle -\nabla f(x^*), t \rangle \leq 0$, which shows that $-\nabla f(x^*) \in \mathbf{N}_{\mathbf{X}}(x^*)$. \square

When the function $f(\cdot)$ is only locally Lipschitz continuous, we obtain following corollary which follows directly from Theorem 5.6 and equation (24a).

Corollary 5.13 *Suppose that the assumptions of Theorem 5.12 are satisfied, but $f(\cdot)$ were only locally Lipschitz continuous. Then,*

$$d^0 f(x^*; d) \geq 0, \quad \forall d \in D_{x^*}(x^*). \quad (25)$$

\square

6 Conclusion

We have extended the family of GPS algorithms to a form that converges to a stationary point of a smooth cost function that cannot be evaluated exactly, but that can be approximated by a family of possibly discontinuous functions $\{f_N(\cdot)\}_{N \in \mathbb{N}^q}$. An important feature of our algorithms is that they use low-cost, coarse precision approximations to the cost function when far from a solution, with the precision progressively increased as a solution is approached. This feature is known to lead to considerable time savings over using very high precision approximations to the cost function in all iterations.

In constructing our algorithms, we have adopted a geometric framework that should be easier to grasp than that found in earlier versions of GPS algorithms.

Appendix

A Extension of Model GPS Algorithm 4.10

We present a modified version of Model GPS Algorithm 4.10 that updates the mesh divider in the same manner as Audet and Dennis [1]. In particular, it allows increasing the mesh divider if the approximate cost function was reduced in the current iteration.

Algorithm A.1 (Model GPS Algorithm that allows increasing the Mesh Size)

- Data:** Initial iterate $x_0 \in \mathbf{X}$;
 Constants $\tau > 1$, $\tau \in \mathbb{Q}$ and $t_{max} \geq 1$, $t_{max} \in \mathbb{N}$;
 Initial mesh divider $\Delta_0 > 0$, arbitrary.
 Base direction matrix $\widehat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S}$ (see Definition 4.2).
- Maps:** As in Model GPS Algorithm 4.10.
- Step 0:** Initialize $k = 0$ and $s_0 = 0$.
- Step 1:** As in Model GPS Algorithm 4.10.
- Step 2:** As in Model GPS Algorithm 4.10.
- Step 3:** Parameter Update
 If there exists an $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ satisfying $f_N(x') < f_N(x_k)$,
 set $x_{k+1} = x'$, $s_{k+1} = s_k + t_k$, with $t_k \in \{t \in \mathbb{N} \mid 0 \leq t \leq t_{max}\}$ arbitrary;
 else, set $x_{k+1} = x_k$, $s_{k+1} = s_k + t_k$,
 with $t_k \in \{t \in \mathbb{N} \mid -t_{max} \leq t \leq -1\}$ arbitrary.
 Set $\Delta_{k+1} = \tau^{s_{k+1}} \Delta_0$ and $N = \rho(\min(\Delta_{k+1}/\Delta_0, 1))$.
- Step 4:** As in Model GPS Algorithm 4.10.
-

Remark A.2 To ensure that $N \geq N_0 = \rho(1)$ during the optimization, in Step 3 we take the minimum of Δ_{k+1}/Δ_0 and 1 as the argument of $\rho(\cdot)$. If Δ_{k+1} is set larger than Δ_0 , then the number of solver iterations may become unreasonable small. \square

Audet and Dennis show that with the mesh construction in Algorithm A.1, the mesh dividers satisfy $\liminf_{k \rightarrow \infty} \Delta_k = 0$. Thus, Algorithm A.1 also constructs a refining subsequence, \mathbf{K} . Furthermore, Step 3 of Algorithm A.1 ensures that

$$\limsup_{k \in \mathbf{K}} \frac{f_N(x_k + \Delta_k d) - f_N(x_k)}{\Delta_k} \geq 0, \quad (26)$$

as we also had in Master GPS Algorithm 4.10.

Consequently, our Theorem 5.6 also holds for Algorithm A.1, and so do Theorem 5.8, Theorem 5.12 and Corollary 5.13.

B Error Bound Functions

We will now show how the error bound function $\varphi: \mathbb{N}^p \rightarrow \mathbb{R}_+$ arises in a few specific optimization problems.

Example B.1 Consider the problem

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{subject to} \quad (27a)$$

$$f(x) \triangleq F(z(x, 1), x), \quad (27b)$$

$$\frac{dz(x, t)}{dt} = y, \quad z(x, 0) = \zeta(x), \quad t \in [0, 1], \quad (27c)$$

$$u(y, x, t) = 0, \quad (27d)$$

where $F: \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ and $u: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ are continuously differentiable. Let y be defined as the solution of (27d), and assume that, for all $x \in \mathbb{R}^n$ and for all $t \in \mathbb{R}$, y is unique and continuously differentiable, but can only be approximated by K iterations of a solver, and denote the approximation by y_K .

We assume that there exist a constant $C_s \in (0, \infty)$ and a known function $\varphi_s: \mathbb{N} \rightarrow \mathbb{R}_+$, satisfying $\varphi_s(K) \rightarrow 0$ as $K \rightarrow \infty$, such that for all x in any compact subset of \mathbb{R}^n , and for all $t \in [0, 1]$,

$$|y - y_K| \leq C_s \varphi_s(K). \quad (27e)$$

In example, when the bisection rule is used for finding y_K , $\varphi_s(K)$ can be taken to be $1/2^K$.

We will use the Euler integration method with $N \in \mathbb{N}$ integration steps. Let $z_N(x, 1)$ denote the numerical solution obtained by solving (27d) with infinite precision. Then it follows from the error analysis of the Euler method [13] that there exist an $N^* \in \mathbb{N}$ and a $C_e > 0$ such that for all $N \in \mathbb{N}$, with $N > N^*$, and for all x belonging to a compact subset of \mathbb{R}^n ,

$$|z_N(x, 1) - z(x, 1)| \leq \frac{C_e}{N}. \quad (27f)$$

Therefore, if $f_{K,N}(\cdot)$ is the cost function associated with the approximate solutions of (27c) and (27d), then there exists a constant $C \in (0, \infty)$ such that for all $N \in \mathbb{N}$, with $N > N^*$, and for all x belonging to a compact subset of \mathbb{R}^n ,

$$|f(x) - f_{K,N}(x)| \leq C \left(C_s \left(\frac{1}{2} \right)^K + C_e \frac{1}{N} \right). \quad (27g)$$

Thus, $\varphi(K, N) = \alpha 2^{-K} + N^{-1}$ for some $\alpha > 0$ and sufficiently large N . \square

Next, we present an example where the cost function is defined on the solution of a partial differential equation, and the boundary condition of the PDE can only be approximated.

Example B.2 Consider the optimization problem of achieving a prescribed temperature profile in a 3-dimensional body at time $t = 1$ by controlling the heat transfer at the body's surface. Components of the design parameter x could be, for example, the nominal heating power and some control parameters.

Let Ω be an open, connected, bounded subset of \mathbb{R}^3 , and let $v_d: \Omega \rightarrow \mathbb{R}$ be given and continuously differentiable. Then the problem can be stated as

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{subject to} \quad (28a)$$

$$f(x) \triangleq \int_{\Omega} |v(x, \zeta, 1) - v_d(\zeta)| \, d\Omega, \quad (28b)$$

$$\nabla_{\zeta}^2 v(x, \zeta, t) = \frac{\partial v(x, \zeta, t)}{\partial t}, \quad \zeta \in \Omega, \quad t \in [0, 1], \quad (28c)$$

$$v(x, \zeta, 0) = v_0(x, \zeta), \quad \zeta \in \Omega,$$

$$\begin{aligned} \nabla_{\zeta} v(x, \zeta, t) \Big|_{\zeta \in \partial\Omega} &= y n(\zeta) \Big|_{\zeta \in \partial\Omega}, \quad t \in [0, 1], \\ g(y, x, t) &= 0, \end{aligned} \quad (28d)$$

where $n(\zeta) \in \mathbb{R}^3$ is the unit normal vector at the boundary points $\zeta \in \partial\Omega$, $\nabla_{\zeta}(\cdot)$ is the gradient with respect to ζ , and $\nabla_{\zeta}^2(\cdot)$ is the Laplacian operator with respect to ζ . Assume that $v_0: \mathbb{R}^n \times \mathbb{R}^3 \rightarrow \mathbb{R}$, and $g: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ are continuously differentiable, and that, for all x and t , (28d) has a unique continuously differentiable solution, but the solution of (28d) can only be approximated.

Let y_K denote the approximate solution of (28d). Assume that y_K satisfies, for some constant $C_s \in (0, \infty)$, and some known function $\varphi_s: \mathbb{N} \rightarrow \mathbb{R}_+$, satisfying $\varphi_s(K) \rightarrow 0$ as $K \rightarrow \infty$,

$$|y - y_K| \leq C_s \varphi_s(K), \quad (28e)$$

for all x belonging to a compact subset of \mathbb{R}^n , and for all $t \in [0, 1]$. (See Example B.1 for a specific error bound function $\varphi_s(\cdot)$.)

Let $v_K(x, \cdot, \cdot)$ denote the infinite precision solution of the PDE (28c) but with finite precision boundary condition

$$\nabla_{\zeta} v_K(x, \zeta, t) \Big|_{\zeta \in \partial\Omega} = y_K n(\zeta) \Big|_{\zeta \in \partial\Omega}, \quad t \in [0, 1], \quad (28f)$$

i.e., y_K is the approximate solution of $0 = g(y, x, t)$.

Then, by linearity of the PDE, the difference $s_K(x, \zeta, t) \triangleq v(x, \zeta, t) - v_K(x, \zeta, t)$ is the solution of the equation

$$\begin{aligned} \nabla_{\zeta}^2 s_K(x, \zeta, t) &= \frac{\partial s_K(x, \zeta, t)}{\partial t}, \quad \zeta \in \Omega, \quad t \in [0, 1], \\ s_K(x, \zeta, 0) &= 0, \quad \zeta \in \Omega, \\ \nabla_{\zeta} s_K(x, \zeta, t) \Big|_{\zeta \in \partial\Omega} &= (y - y_K) n(\zeta) \Big|_{\zeta \in \partial\Omega}, \quad t \in [0, 1]. \end{aligned} \quad (28g)$$

For any function $g: \Omega \times \mathbb{R} \rightarrow \mathbb{R}$, let $\|g(\cdot, \cdot)\|_\infty \triangleq \sup_{\zeta \in \Omega, t \in [0, 1]} |g(\zeta, t)|$. Then, (28e) together with the linearity of the PDE implies that there exists a constant $C_l \in (0, \infty)$ such that

$$\|s_K(x, \cdot, \cdot)\|_\infty \leq C_l \varphi_s(K). \quad (28h)$$

Let $M \in \mathbb{N}$ be the number of mesh points for each coordinate direction, used for the spatial discretization, and let $N \in \mathbb{N}$ be the number of mesh points for the temporal discretization. For given $M, N \in \mathbb{N}$, and $x \in \mathbb{R}^n$, let $\{v_{M,N}(x, \cdot, \cdot)\}$ be the approximate solutions of (28c) (subject to the infinite precision boundary condition), and, for given $K \in \mathbb{N}$, let $\{v_{K,M,N}(x, \cdot, \cdot)\}$ be the approximate solutions of (28c) (subject to the approximate boundary condition (28f)). Similarly, let $\{s_{K,M,N}(x, \cdot, \cdot)\}$ be the approximate solutions of (28g). Suppose that the integration scheme is stable and such that there exist constants $C_I \in (0, \infty)$, $p > 1$ and $q > 1$, such that for all x belonging to a compact subset of \mathbb{R}^n and for all sufficiently large $M, N \in \mathbb{N}$,

$$\|v(x, \cdot, \cdot) - v_{M,N}(x, \cdot, \cdot)\|_\infty \leq C_I (M^{-p} + N^{-q}), \quad (28i)$$

$$\|s_K(x, \cdot, \cdot) - s_{K,M,N}(x, \cdot, \cdot)\|_\infty \leq C_I (M^{-p} + N^{-q}). \quad (28j)$$

Then,

$$\begin{aligned} \|v(x, \cdot, \cdot) - v_{K,M,N}(x, \cdot, \cdot)\|_\infty &\leq \|v(x, \cdot, \cdot) - v_{M,N}(x, \cdot, \cdot)\|_\infty \\ &\quad + \|v_{M,N}(x, \cdot, \cdot) - v_{K,M,N}(x, \cdot, \cdot)\|_\infty \\ &\leq C_I (M^{-p} + N^{-q}) + \|s_{K,M,N}(x, \cdot, \cdot)\|_\infty \\ &\leq C_I (M^{-p} + N^{-q}) + C_l \varphi_s(K) + C_I (M^{-p} + N^{-q}) \\ &\leq C (M^{-p} + N^{-q} + \alpha \varphi_s(K)) \end{aligned} \quad (28k)$$

for some $\alpha, C \in (0, \infty)$.

Thus, for some $C' \in (0, \infty)$ and sufficiently large $M, N \in \mathbb{N}$, $|f(x) - f_{K,M,N}(x)| \leq C' (M^{-p} + N^{-q} + \alpha \varphi_s(K))$, and $\varphi(M, N, K) = M^{-p} + N^{-q} + \alpha \varphi_s(K)$. \square

Example B.3 In [11], Pironneau and Polak present a two-point boundary value optimal control problem with scalar, linear double integrator dynamics which they approximate using the finite difference method. The resulting finite difference equation is then solved using the Gauss-Seidel method. If $K \in \mathbb{N}$ is the number of discretization steps, and $N \in \mathbb{N}$ is the number of Gauss-Seidel iterations, then the error bound for the cost function is shown to be

$$\varphi(K, N) = \left(1 - c \left(\frac{1}{K}\right)^2\right)^N \quad (29a)$$

where $c \in (0, 1)$ is an unknown constant. The constant c can be guessed, or one can replace the function $\varphi(\cdot, \cdot)$ with the conservative estimate $\varphi(K, N) = (1 - K^{-(2+\epsilon)})^N$, with $0 < \epsilon \ll 1$, small, i.e., replace c with $K^{-\epsilon}$. To ensure that $\varphi(K, N) \rightarrow 0$, as $K, N \rightarrow \infty$, we set $N(K)$ to the smallest integer such that

$$N(K) \geq C K^{2+2\epsilon}, \quad (29b)$$

with $C > 0$ arbitrary. Then,

$$N(K) \approx C K^{2+2\epsilon}. \quad (29c)$$

Note that

$$\begin{aligned} (1 - K^{-(2+\epsilon)})^{C K^{2+2\epsilon}} &= \exp\left(C K^{2+2\epsilon} \log(1 - K^{-(2+\epsilon)})\right) \\ &\approx \exp(-C K^\epsilon) \rightarrow 0, \text{ as } K \rightarrow \infty. \end{aligned} \quad (29d)$$

□

References

- [1] Charles Audet and J. E. Dennis, Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2003.
- [2] Andrew J. Booker, J. E. Dennis, Jr., Paul D. Frank, David B. Serafini, Virginia Torczon, and Michael W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13, February 1999.
- [3] F. H. Clarke. *Optimization and nonsmooth analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.
- [4] I. D. Coope and C. J. Price. Positive bases in numerical optimization. Technical Report UCDMS2000/12, Dept. of Mathematics and Statistics, Univ. of Canterbury, Christchurch, New Zealand, 2000.
- [5] Chandler Davis. Theory of positive linear dependence. *American Journal of Mathematics*, 76(4):733–746, October 1954.
- [6] J. E. Dennis, Jr. and Virginia Torczon. Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1(4):448–474, 1991.
- [7] J. E. Dennis, Jr. and Virginia Torczon. Managing approximation models in optimization. In Natalia M. Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, ICASE/NASA Langley Workshop on Multidisciplinary Optimization, pages 330–347. SIAM, 1997.
- [8] R. Hooke and T. A. Jeeves. 'Direct search' solution of numerical and statistical problems. *J. Assoc. Comp. Mach.*, 8(2):212–229, April 1961.
- [9] Robert Michael Lewis and Virginia Torczon. Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization*, 9(4):1082–1099, 1999.
- [10] Robert Michael Lewis and Virginia Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000.

- [11] Olivier Pironneau and Elijah Polak. Consistent approximations and approximate functions and gradients in optimal control. Technical Report UCB/ERL M00/14, University of California at Berkeley, Electronics Research Laboratory, March 2000. To appear in SIAM Journal on Control and Optimization.
- [12] Elijah Polak. *Computational Methods in Optimization; a Unified Approach*, volume 77 of *Mathematics in Science and Engineering*. New York, Academic Press, 1971.
- [13] Elijah Polak. *Optimization, Algorithms and Consistent Approximations*, volume 124 of *Applied Mathematical Sciences*. Springer Verlag, 1997.
- [14] David B. Serafini. *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*. PhD thesis, Rice University, 1998.
- [15] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.
- [16] Virginia Torczon and Michael W. Trosset. Using approximations to accelerate engineering design optimization. *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, AIAA Paper 98-4800*, September 1998.
- [17] Michael W. Trosset. On the use of direct search methods for stochastic optimization. Technical Report TR00-20, Rice University, Department of Computational and Applied Mathematics, 2000.